IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

Applicant	: M. Hamburg and A.M. Herasimchuk	Art Unit	: 2771
		Examiner	: Ella Colbert
Serial No.	: 09/010,801		
Filed	: January 22, 1998		
Title	: MAINTAINING DOCUMENT STATE HISTORY		

BOX AF
Commissioner for Patents
Washington, D.C. 20231

<u>BRIEF ON APPEAL</u>

(1)   Real Party in Interest

The real party in interest is Adobe Systems Incorporated.

(2)   Related Appeals and Interferences

None.

(3)   Status of Claims

Claims 3, 4, 8, 37 and 38 stand rejected under 35 U.S.C. § 112, par. 2.

Claims 1-18, 22-25 and 37-44 stand rejected under 35 U.S.C. § 103(a) as unpatentable over U.S. Patent No. 5,659,747 ("Nakajima").

Claims 19-21 stand rejected under 35 U.S.C. § 103(a) as unpatentable over Nakajima in view of U.S. Patent No. 6,018,342 ("Bristor").

Claims 27-36, 44 stand rejected under 35 U.S.C. § 103(a) as unpatentable over Najakima in view of U.S. Patent No. 5,583,982 ("Matheny").

Claim 26 has been cancelled.

(4) Status of Amendments

There are no unentered amendments.

(5) Summary of Invention

Methods and apparatus embodying techniques useful in systems for creating and modifying documents. A state history of a document for storing document states is maintained and, whenever an operation of a predetermined type has occurred, the state of the document is automatically captured as it exists after the operation. The captured state is added to the state history. (Specification, page 2, lines 11-19)

In another aspect, the techniques identify for a user on a display device a set of states that the document has been in by operation of the system and enable the user to designate any one of the identified states for further operations. The techniques may provide both linear and non-linear history. (Specification, page 4, lines 1-5, 25-28)

Another aspect of the invention includes techniques to maintain a first history of interesting operations and a second history of all operations requested by a user, the second history but not the first history including operations global to the state of the application. (Specification, page 2, lines 20-25)

In another aspect, the techniques include operations to keep a history of document states created by a user; enable the user to discard any of the history; and enable the user to step backward and forward through the history and thereby to alter the state of the document to be any of the document states in the history. (Specification, page 3, lines 17-22)

In another aspect, the techniques include operations to keep a history list; go back to a previous state in the history list; select a future state from the history list, being a state created after the previous state, as a source of data for an operation; and perform the operation with the future data on the previous state. (Specification, page 3, lines 10-16)

(6) Issues

1. Are claims 3, 4, 8, 37 and 38 properly rejected under 35 U.S.C. § 112, par. 2?

2. Are claims 1-44 properly rejected under 35 U.S.C. § 103(a) as unpatentable over Nakajima or Nakajima in view of Bristor or Matheny?

(7)   **Grouping of Claims**

For the purpose of this appeal, the claims are grouped as follows:

1.   For the purpose of Issue 1, claims 3, 4, 8, 37 and 38 stand and fall together.

2.   For the purpose of Issue 2:

    a. Claims 1-7, 9-15, 17-21, 37, 39, 41 and 42 stand together.

        i. Claim 14 is separately patentable from claims listed in (a) above.

        ii. Claims 17, 18-21, 41 and 42 are separately patentable from claims listed in (a) above.

            1. Claims 18-21 and 42 are separately patentable from claims listed in (ii) above.

    b. Claims 8 and 38 stand or fall together.

    c. Claims 16 and 40 stand or fall together.

    d. Claims 22-25, 27-36, 43 and 44 stand or fall together.

(8)   **Argument**

1.   **Are claims 3, 4, 8, 37 and 38 properly rejected under 35 U.S.C. § 112, par. 2?**

Claims 3, 4, 8, 37 and 38 have been rejected for having an insufficient antecedent basis for the limitation "interesting operation". Claim 3 in fact defines the limitation "interesting operation", wherein it recites: "an operation is classified as an interesting operation if it changes the state of the document" (Claim 3, lines 7-8, page 20). Claim 3 establishes a basis for reference of this limitation in dependant claim 4.

Additionally, "interesting operation" is defined in the specification at page 8, lines 14-16, as a command that changes the document state, thus providing a basis for reference of this limitation in claims 8, 37 and 38.

2.   Are claims 1-44 properly rejected under 35 U.S.C. § 103(a) as unpatentable over Nakajima or Nakajima in view of Bristor or Matheny?

    Group (a): Claims 1-7, 9-15, 17-21, 37, 39, 41 and 42

Claim 1 recites a method for maintaining in a memory a state history of a document, automatically capturing the state of the document as it exists after an operation of a predetermined type has occurred, and adding that captured state to the state history. Claim 3 recites a method that includes performing a "step backward operation" or a "step forward operation" by installing as the current state of the document a state stored in the state history that was created either before or after the current state. The balance of the claims in Group (a) incorporate either or both of these features.

Nakajima does not teach or suggest a method for maintaining a state history. Nakajima teaches a fundamentally different method. Nakajima teaches maintaining only a single state of a document in conjunction with a log of user commands that have been applied to the document. The data held in the command log and functions for maintaining the data are encapsulated into a command log object (col. 3, lines 21-24). This command log object knows how to do, undo and redo an operation (col. 3, lines 35-41). The mechanism maintains a current position in the log to monitor a current state of the application program (col. 2, lines 40-42).

The mechanism for performing multiple-level undo/redo operations taught by Nakajima allows a single state of a document (*i.e.,* the current state) to be changed in response to user commands, some of which (*i.e.,* undo commands) may result in the effects of a previously executed command to be reversed, thereby reproducing an earlier state of the document. As such, the undo command "restores" a document to a state it had before the operation being undone was done, by performing an inverse operation on the current state of the document, rather than restoring a saved, previous state of the document, as would be the case if Applicant's invention were applied. Nakajima does not teach a method of maintaining in memory an earlier state of a document. Such an earlier state may be returned to directly.

The Nakajima method of multiple-level undo/redo operations is similarly taught in Gamma et al., Design Patterns, Addison-Wesley Publishing Co., © 1995, pages 233 to 242, as discussed within Applicant's specification (line 18, page 1; line 9, page 2).

Nakajima does not teach a step backward or step forward operation that installs as the current state of the document a previously stored state. The Examiner conceded that Nakajima did not teach "performing a step forward operation", but argued that Nakajima's performing a redo operation is synonymous to performing a step forward operation. Nakajima does not teach maintaining more than a single state of a document. Rather, Nakajima's mechanism for performing a redo operation merely changes the current state of the document by performing a command, the effects of which command were previously reversed by an undo command, and it does not retrieve from memory a "future", saved state of the document, and install that future, saved state as the current state. Contrary to the Examiner's unsupported factual assertion, the redo operation of Nakajima is not "synonymous" to the step forward operation, but is fundamentally distinct.

Similarly, the Examiner conceded that Nakajima did not teach a "step backward operation", but argued, with no factual support, that Nakajima's performing an undo operation is "synonymous" with a step backward operation. For the same reasons stated above in relation to the step forward operation, Nakajima's undo operation is not synonymous with the step backward operation. Nakajima's undo operation merely changes the current state of the document by performing a command that results in reversing the effects of a previously executed command, and does not retrieve from memory a saved, previous state of the document and install it as the current document – as would be the case were Applicant's invention to be applied.

Because at least these elements of Applicant's claimed invention are not taught or suggested by Nakajima, no *prima facie* case of obviousness under 35 U.S.C. § 103 has been established.

Sub-Group (a)(i): Claim 14

Claim 14 is separately patentable from the balance of the claims in Group (a), as it recites the additional feature of a non-linear history mode. This feature has the advantage of allowing a user to edit a document starting from an earlier-captured document state without losing later-captured states (Specification, page 4, lines 25-28). For example, using a digital painting application, a user may apply a series of filters to an image, then go back to the original image

state, and then selectively paint in the effects of those filters using a history paintbrush tool (Specification, page 5, lines 28-30, page 6, lines 1-2).

Claim 14 recites a computer-implemented method of interacting with a user editing a document in a computer program application that incorporates the features of claim 9 in addition to a feature wherein a command to change the document, that comes after a step backward command to a selected item in the history list, does not cause the items after the selected item to be deleted from the history list, and adds a new item to the end of the history list and a new document state to the state history. Nakajima does not teach the non-linear history mode recited in claim 14. If Nakajima's command log is considered akin to the history list (which the Applicant denies), then Nakajima does not permit, and in fact teaches away from, the feature recited in claim 14. In column 4, lines 44-58, Nakajima teaches that previously undone commands in the command log are deleted as soon as the user enters a new command, other than an undo/redo command. Therefore, Applicant submits that claim 14 is allowable over Nakajima.

Sub-Group (a)(ii): Claims 17, 18-21, 41 and 42

Claims 17, 18-21, 41 and 42 are separately patentable from the balance of the claims in Group (a). Claims 17 and 18 recite a method to edit a document keeping a history of document states created by a user, enabling the user to discard any of the history and to step backward and forward through the history, thereby altering the state of the document to be any of the document states in the history. The balance of the claims in this sub-group incorporate this feature.

As discussed above, Nakajima teaches maintaining only a single state of a document, in conjunction with a log of user commands that have been applied to the document. Nakajima does not teach a method of maintaining in memory more than one state of a document, enabling a user to step forward and backward through the state history. Further, Nakajima does not permit a user to discard *any* previous user command.

Nakajima maintains a sequential log of user commands in memory in a sequence ranging from a selected user command that has been executed to a most recently executed user command (col. 1, lines 43-54). Even if it were possible in Nakajima to remove *any* command from this sequential list, as recited in the claims, which Nakajima does not even suggest, the effect of doing so on commands coming later in sequence would be unpredictable at best. If a proposed

modification would render the prior art unsatisfactory for its intended purpose, then there is no

suggestion or motivation to make the proposed modification (*In re Gordon*, 733 F.2d 900 (Fed.

Cir. 1984)). For at least this reason the claims in this sub-group are allowable.

Sub-Group (a)(ii)(1): Claims 18-21 and 42

Claims 18-21 and 42 are separately patentable from the balance of the claims in Group

(a) and Sub-Group (a)(ii). In addition to enabling a user to discard any user-selected set of the

document states in the history, they include an additional feature of enabling the user to

designate any one of the document states in the history and thereby install the designated state as

the current state of the document.

Because Nakajima's system does not permit removing a command from the command

log without adverse effect, it follows Nakajima also does not permit enabling a user to discard

any user-selected set of the document states and to install any designated state of the document

states in the history as the current state.   Claims 18-21 and 42 recited these features.

Applicant's system enables a user to do this because the later document states are complete in

themselves and earlier ones can be deleted with no adverse effect. (Specification, page 12, lines

1-3)

For at least this reason the claims in this sub-group are allowable.

Group (b): Claims 8 and 38

Claims 8 and 38 stand or fall separately, as claim 8 recites a method enabling a user to

undo revisions made to a document by maintaining a first history of interesting operations and a

second history of all operations requested by a user, the second history but not the first history

including operations global to the state of the application, which feature is not recited in the

balance of the claims. Claim 38 is an apparatus claim that recites corresponding features.

Nakajima does not disclose maintaining two distinct histories: a first history of interesting

operations and a second history of operations global to the state of the application. In column 1,

lines 29-34 and lines 39-48, Nakajima teaches a method of maintaining a single log or history of

user commands to perform a single undo/redo operation, multiple successive undo/redo

operations or complete undo/redo operations (abstract, see also col. 1, lines 31-32, lines 36-38,

lines 43-45). Nakajima neither teaches nor suggests making any distinction as to how different kinds of operations are handled, let alone the distinction recited in claim 8. Therefore, Applicant submits that claims 8 and 38 are allowable over Nakajima.

Group (c): Claims 16 and 40

Claims 16 and 40 stand or fall separately, as claim 16 recites a method for editing a document, whereby a previous state of the document, maintained in the history list, is returned to and a future state of the document from the history list is selected as a source of data for performing an operation with said future data on the previous state, which is not disclosed by the balance of the claims. Claim 40 is a computer program product claim incorporating corresponding features. These claims are therefore patentable independent of the patentability of the balance of the claims.

Nakajima does not teach using saved, future states of a document as a source of data to perform an operation on a different, previous state of the document. In column 5, lines 15-24 and 47-65, Nakajima merely teaches performing multiple undo operations, whereby the current state of a document is changed to reverse the effects of previously executed commands, to in effect reproduce an earlier, unsaved state of the document. Nakajima does not permit using data from a future, saved state of the document to perform an operation on a previous, saved state of the document, since Nakajima teaches maintaining only one state of the document in memory. Applicant submits that claims 16 and 40 are therefore allowable over Nakajima.

Group (d): Claims 22-25, 27-36, 43 and 44

Claim 22 recites a method of enabling a user to control operation of a computer program application for creating and modifying a document. The method includes identifying for the user on a display device a set of states that the document has been in by operation of the application, enabling the user to designate *any* one of the identified states, and providing the user an editing tool having the designated state as a document state operand. Nakajima only permits a user to use an editing tool to give editing commands to the current state of a document, since Nakajima does not teach saving previous and future states of a document in a state history. As such, in Nakajima there are no other states for a user to designate as the document state operand.

Further, Matheny does not provide the user an editing tool having the designated state as a document state operand as required by claim 22. An example of such a tool is the history paintbrush tool described in the Applicant's specification on page 4, lines 28-30, through page 5, lines 1-2. Matheny provides generic commands for Cut, Copy, paste, Starting HyperMedia links, Pulling Data on Links, as well as other user interface commands (col. 5, lines 30-33). In column 13, lines 7-33, Matheny teaches a standard editing protocol. The Matheny standard editing protocol supports standard editing commands such as Cut, Copy, Paste, Push Data, etc. (col. 13, lines 8-13).

Column 5, lines 37-40 of Matheny teaches command objects for encapsulating all of the information necessary to undo a command after a command is done. Matheny does not teach providing a user with a delete tool for deleting a designated state from a set of states. Matheny does teach using an undo command to reverse the effects of a previously executed command, but does not teach deleting the state of the document upon which said commands are executed.

Nakajima in combination with Matheny does not teach or suggest providing the user the editing method recited in claim 22. For at least these reasons, claim 22 is allowable. Claims 23-25, 27-36, 43 and 44 incorporate the features of claim 22 and are allowable for at least the same reasons set forth above in connection with claim 22.

(9)   Appendix

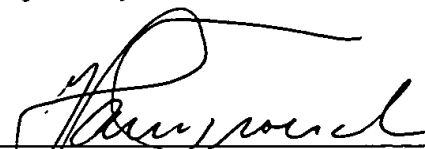Appendix A to this brief is a set of the claims currently pending in this case.

Applicant : M. Hamburg and A.M. Herasimchuk   Attorney's Docket No.: 07844-235001
Serial No. : 09/010,801
Filed   : January 22, 1998
Page   : 10

Please apply the brief fee of $310 and any other charges or credits to Deposit Account No. 06-1050, making reference to attorney docket no. 07844-235001.

Respectfully submitted,

Date: 20 Feb, 01

Hans R. Troesch
Reg. No. 36,950

Fish & Richardson P.C.
2200 Sand Hill Road, Suite 100
Menlo Park, CA 94025
Telephone: (650) 322-5070
Facsimile: (650) 854-0875

50037986.doc

APPENDIX A

Pending Claims for 09/010,801, as of 2/20/01

1. A method implemented in a computer program application performing operations on documents having states, the method comprising:

maintaining in a memory a state history of a document for storing document states; and

whenever an operation of a predetermined type has occurred, automatically capturing the state of the document as it exists after the operation and adding the captured state to the state history.

2. The method of claim 1, wherein the memory comprises a disk file.

3. The method of claim 1, wherein:

the state history includes states of the document and the order in which the stored states were automatically added to the state history;

the state history is displayed to a user as a list of document states shown in their stored order;

an operation is classified as an interesting operation if it changes the state of the document;

a state is added to the state history only if the operation creating the state is classified as an interesting operation and not otherwise;

performing a step backward operation by installing as the current state of the document a state stored in the state history, whereby all step backward operations place the document in a state that occurred immediately after an interesting operation;

performing a step forward operation by installing as the current state of the document a state stored in the state history, whereby all step forward operations place the document in a state that occurred immediately after an interesting operation;

4. The method of claim 3, wherein:

the list of document states displayed to the user comprises a list of items, each item representing a state of the document that existed after an interesting operation and that can be recovered with a step backward operation in the application; and

the list of document states displayed to the user comprises a list of items, each item representing a state of the document that existed after an interesting operation and that can be recovered directly by selecting the item.

5. The method of claim 4, wherein:

the application is digital graphics program operable to create and revise images in digital form;

the application provides a tool operable under user control to obtain source material from any state in the state history and apply it to a current state of the document; and

the images are raster images.

6. The method of claim 4, wherein:

the application enables a user to select any item in the displayed list of items and cause the application to create a new document having the document state corresponding to the selected item.

7. The method of claim 4, wherein:

each of the captured states in the state history maintains the state data in essentially its original form, whereby the captured state data is suitable for immediate use in other operations.

8. A method for enabling a user to undo revisions made to a document, the method comprising:

maintaining a first history of interesting operations and a second history of all operations requested by a user, the second history but not the first history including operations global to the state of the application.

9. A computer-implemented method of interacting with a user editing a document in a computer program application, the document having a document state, the method comprising:

receiving from the user a sequence of commands to change the document;

changing the document state in response to each command;

adding the changed document state to a state history maintained in a computer-readable memory device each time the document state is changed;

for each document state added to the state history, adding a corresponding entry to a history list displayed to the user on a computer-controlled display device operated as part of a graphical user interface; and

in response to a user action stepping backward to an item in the history list, updating the document to have the corresponding document state saved in the state history.

10. The method of claim 9, wherein:

the state history and the history list are limited to storing a preset number of items and excess items are scrolled off the top of the list as new items are added.

11. The method of claim 9, wherein:

the state history is stored in a region of memory and the oldest document states in the state history are discarded when free space in the region runs low.

12. The method of claim 11, wherein:

the oldest document states are found and discarded by a memory management process.

13. The method of claim 9, wherein:

a command to change the document that comes after a step backward command to a selected item in the history list causes the items after the selected item to be deleted from the history list and the corresponding document states to be deleted from the state history.

14. The method of claim 9, wherein:

a command to change the document that comes after a step backward command to a selected item in the history list does not cause the items after the selected item to be deleted from the history list and adds a new item to the end of the history list and a new document state to the state history.

15. The method of claim 9, further comprising:

enabling a user interface gesture on the history list to create a new document from a document state from the state history.

16. A method implemented in a computer program application operable to create and edit a document, comprising:

keeping a history list;

going back to a previous state in the history list;

selecting a future state from the history list, being a state created after the previous state, as a source of data for an operation; and

performing the operation with the future data on the previous state.

17. A method implemented in a computer program application operable to create and edit a document, comprising:

keeping a history of document states created by a user;

enabling the user to discard any of the history; and

enabling the user to step backward and forward through the history and thereby to alter the state of the document to be any of the document states in the history.

18. A method implemented in a document-processing computer program application, the method comprising:

keeping a history of document states created automatically whenever a user command to the application changes the state of a document;

enabling the user to discard any user-selected set of the document states in the history; and

enabling the user to designate any one of the document states in the history and thereby install the designated state as the current state of the document.

19. The method of claim 18, further comprising:

saving the history when the document is closed on a long-term storage medium, whereby the history may be restored when the document is later opened and across invocations of the application.

20. The method of claim 19, wherein:

the saved history resides in the document with final document data.

21. The method of claim 19, wherein:

the saved history resides in a long-term data repository independent of the original document.

22. A method enabling a user to control operation of a computer program application for creating and modifying a document, the method comprising:

identifying for the user on a display device a set of states that the document has been in by operation of the application;

enabling the user to designate any one of the identified states; and

providing the user an editing tool having the designated state as a document state operand.

23. The method of claim 22, further comprising:

displaying the document in a user interface window, the document being a digital image.

24. The method of claim 23, wherein the digital image has a plurality of layers, each of the plurality of layers having a plurality of channels, the method further comprising:

displaying user-interface elements for applying filters to the digital image.

25. The method of claim 22, further comprising:

installing the designated state as the current state of the document in response to a user command.

26.   Canceled.

27. The method of claim 22, further comprising:

providing the user an delete tool for deleting the designated state from the set of states.

28. The method of claim 22, wherein:

the set of states is identified by displaying a scrollable list of elements each identifying one of the states in the set.

29. The method of claim 28 wherein the list elements are ordered by the time the corresponding states were created.

30. The method of claim 25 wherein the designation and installation are performed in response to a single command.

31. The method of claim 25 wherein the set of states is displayed in an order, the method further comprising:

enabling the user to make a gesture on a user interface indicating a sequence of displayed state identifiers and responding to the gesture by displaying the document in the states indicated as the gesture is made.

32. The method of claim 25 further comprising:

enabling the user to modify the document state after the installing step; and

adding the document state resulting from the modification to the set of states identified on the display device.

33. The method of claim 31 wherein the set of states is displayed in order of creation of the states in the set.

34. The method of claim 31 wherein the document is a digital image.

35. The method of claim 25 further comprising:

providing a step backward and a step forward command for the user to execute to navigate the set of states; and

providing a separate undo and redo command for the user to undo and redo commands entered by the user.

36. The method of claim 22, further comprising:

providing a step backward and a step forward command for the user to execute to navigate the set of states; and

providing a separate undo and redo command for the user to undo and redo commands entered by the user.

37. Apparatus comprising a computer-readable storage medium tangibly embodying program instructions defining a computer program application for performing operations on documents having states, the program comprising instructions operable for causing a programmable processor to:

maintain in a memory a state history of a document for storing document states; and

whenever an interesting operation has occurred, automatically capture the state of the document as it exists after the operation and adding the captured state to the state history.

38. Apparatus comprising a computer-readable storage medium tangibly embodying program instructions for use by a user of a program to undo revisions made to a document, the apparatus comprising instructions operable for causing a programmable processor to:

maintain a first history of interesting operations and a second history of all operations requested by a user, the second history but not the first history including operations global to the state of the application.

39. Apparatus comprising a computer-readable storage medium tangibly embodying program instructions for interacting with a user editing a document in a computer program application, the document having a document state, the apparatus comprising instructions operable for causing a programmable processor to:

receive from the user a sequence of commands to change the document;

change the document state in response to each command;

add the changed document state to a state history maintained in a computer-readable memory device each time the document state is changed;

for each document state added to the state history, add a corresponding entry to a history list displayed to the user on a computer-controlled display device operated as part of a graphical user interface; and

in response to a user action stepping backward to an item in the history list, update the document to have the corresponding document state saved in the state history.

40. A computer program, residing on a computer-readable medium, comprising instructions for causing a computer to:

keep a history list;

go back to a previous state in the history list;

select a future state from the history list, being a state created after the previous state, as a source of data for an operation; and

perform the operation with the future data on the previous state.

41. A computer program, residing on a computer-readable medium, comprising instructions for causing a computer to:

keep a history of document states created by a user;

enable the user to discard any of the history; and

enable the user to step backward and forward through the history and thereby to alter the state of the document to be any of the document states in the history.

42. A computer program, residing on a computer-readable medium, comprising instructions for causing a computer to:

keep a history of document states created automatically whenever a user command to the application changes the state of a document;

enable the user to discard any user-selected set of the document states in the history; and

enable the user to designate any one of the document states in the history and thereby install the designated state as the current state of the document.

43. A computer program, residing on a computer-readable medium, comprising instructions for causing a computer to:

create and modify a document;

identify for a user on a display device a set of states that the document has been in by operation of the application; and

enable the user to designate any one of the identified states.

44.   The method of claim 36, further comprising:

providing to the user a first undo command function that operates with reference to the first history and a second undo command function that operates with reference to the second history.